DataModels

<<class>>
ClientGroup
Refer to the Data Model
+ClientGroup(name:String,locationId:int, connectionBW:int=0): ClientGroup

<<class>>
Location
Refer to DataBaseModel
+Location(name:String): Location

<<class>>
NetworkLink
Refer to DataBase model
+NetworkLink(locationA:int,locationB:int, capacity:int=0): NetworkLink
+toString(): String

<<class>>
MigrationCostMultiplier
Refer to Database model
+Migration(cost(popAId:int,popBId:int,cost:float=1.0): MigrationCost
+toString(): String

<<class>>
Flavor
Refer to DataBaseModel
+Flavor(id:String,popId:int): Flavor
+updateValues(name:String,cpu:int,ram:int, disk:int,isPublic:boolean)

<<class>>
Hypervisor
Refer to DataBaseModel
+Hypervisor(name:String,popId:int): Hypervisor
+updateCurValues(cpu:int,ram:int,disk:int instances:int)
+updateMaxValues(cpu:int,ram:int,disk:int)

<<class>>
Metric
Refer to Database Model
+Metric(instanceId:int): Metric
+updated(duration:int,sampletime:String,karg:Dictionary<float>)

<<class>>
POP
Refer to Database Model
+POP(name:String,locationId:int,url:String, tenant:String,loginUser:String,loginPass:String, region:String,org:None): POP
+updateMaxValues(cpu:int,ram:int,disk:disk, instances:int)
+updateCurValues(cpu:int,ram:int,disk:int, instances:int)
+setTotalValues(cpu:int,ram:int,disk:int, instances:int,netBW:int)
+canhostVCDN(AvCDN:vCDN): boolean

<<class>>
Instances
Refer to DataBase Model
+Instance(vcdnId:int,popId:int): instance
+updateMaxValues(cpu:int,ram:int,instances:int)
+updateCurValues(cpu:int,ram:int,instances:int)

<<class>>
HmacResult
Refer to Database model
+HmacResult(popId:int,instanceId:int, cost:int=0,delay:int=0,minBW:int=0): HmacResult

<<class>>
AlteredDemand
Refer to database model
+AlteredDemand(demandId:int,dstPopId:int): AlteredDemand

<<class>>
Demand
Refer to Database model
+Demand(clientGroupId:int,popId:int,vcdnId:int, volume:int=0,bufTime=0): Demand

<<class>>
vCDN
Refer to data Model
+vCDN(name:String,tenant:String,loginUser:String, loginPass:String): vCDN

sqlalchemy